Characterizing viral quasispecies

Gregori, J and Guerrero, M

09/05/2018

Contents

1 Quasispecies complexity by biodiversity indices

RNA and DNA viruses that replicate by low fidelity polymerases generate a viral quasispecies, a collection of closely related viral genomes.

The high replication error rate that generates this quasispecies is due to a lack of proofreading mechanisms. This feature, in addition to the high replication rate of some viruses, leads to generation of many mutations in each cycle of viral replication.

The complexity of the quasispecies contributes greatly to the adaptive potential of the virus. The complexity of a viral quasispecies can be defined as an intrinsic property that quantifies the diversity and frequency of haplotypes, independently of the population size that contains them. (Gregori et al. 2016) (Gregori et al. 2014)

Quasispecies complexity can explain or predict viral behavior, so it has an obvious interest for clinical purposes. We are often interested in comparing the viral diversity indices between sequential samples from a single patient or between samples from groups of patients. These comparisons can provide information on the patient's clinical progression or the appropriateness of a given treatment.

QSUtils is a package intended for use with quasispecies amplicon NGS data, but it could also be useful for analyzing 16S/18S ribosomal-based metagenomics or tumor genetic diversity by amplicons.

In this tutorial we illustrate the use of functions contained in the package to compute diversity indices, which can be classified into incidence-based, abundance-based, and functional indices.

2 Install package and load data

First, the package should be installed and loaded.

```
library(devtools)
install_git("https://github.com/VHIRHepatiques/QSutils")

library(QSutils)
```

Then, the dataset to work with the diversity indices should be loaded. This vignette deals with the different ways to load data: Simulation vignette for package users is also available.

```
lst <- ReadAmplSeqs("../inst/extdata/ToyData_10_50_1000.fna",type="DNA")
lst
## $nr
##  [1] 464  62  39  27  37  16  33  54 248  20
##
## $hseqs
##   A DNAStringSet instance of length 10
##      width seq                                          names
## [1]     50 CACCCTGTGACCAGTGTGTGC...TGCCCCGTTGGCATTGACTAT Hpl_0_0001|464|46.4
## [2]     50 CACTCTGTGACCAGTGTGTGC...TGCCCCGTTGGCATTGACTAT Hpl_1_0001|62|6.2
## [3]     50 CACCCTGTGACCAGCGTGTGC...TGCCCCGTTGGCATTGACTAT Hpl_1_0002|39|3.9
## [4]     50 CACCCTGTGACCAGTGTGTGC...TGCCCCGTTGGCATTGACTAC Hpl_1_0003|27|2.7
## [5]     50 CACTCTGTGACCAGTGTGTGC...TGCCCCGTTAGCATTGACTAT Hpl_2_0001|37|3.7
```

```
##  [6]     50 CACTCGGTGACCAGTGTGCGT...TGCCCCGTTGGCATTGACTAT Hpl_4_0001|16|1.6
##  [7]     50 CACTCTGTGACCAGTGTGCGC...TGCCCTGTCGGCATTGACTAT Hpl_5_0001|33|3.3
##  [8]     50 CACTCTGTGATCAGTGTGCGC...TGCCCTGCCGGCATCGACTAT Hpl_8_0001|54|5.4
##  [9]     50 CACTCTGTGATCAGTGTGCGC...TGCCCTGCCGGCATCGACTAC Hpl_9_0001|248|24.8
## [10]     50 CACTCTGTGATCAGTGTGCGC...TGCCCTGCCGGCACCGACTAC Hpl_10_0001|20|2
```

3   Incidence-based diversity indices or richness indices

Incidence-based diversity indices correspond to the number of observed entities, regardless of their abundances. The main incidence indices are the number of haplotypes, the number of polymorphic sites, and the number of mutations. These indices can be computed as follows:

- number of haplotypes

```
length(lst$hseqs)
## [1] 10
```

In this fasta file, 10 haplotypes are reported. This can be calculated with the length of the DNAStringSet object.

- The number of polymorphic sites

```
SegSites(lst$hseqs)
## [1] 14
```

In this example, there are 14 polymorphic sites; that is, mutated positions with respect to the dominant haplotype.

- Number of mutations

```
TotalMutations(lst$hseqs)
## [1] 37
```

Returns the total number of mutations observed in the alignment.

4   Abundance-based diversity indices

Abundance-based diversity indices take into account the entities present and their relative abundance in the population.

- Shannon entropy

```
Shannon(lst$nr)
## [1] 1.6396
NormShannon(lst$nr)
## [1] 0.7120691
```

Shannon entropy is a measure of uncertainty. This index varies from 0, when there is a single haplotype, to the logarithm of the number of haplotypes. It represents the uncertainty in assigning a random genome to the corresponding haplotype in the population studied. This index is commonly used in the normalized form, although the non-normalized form is recommended. The normalized form constitutes a measure of evenness rather than diversity.

The variance of Shannon entropy and normalized Shannon entropy can be computed with:

```
ShannonVar(lst$nr)
## [1] 0.001154642
NormShannonVar(lst$nr)
## [1] 0.000217779
```

- Gini-Simpson

```
GiniSimpson(lst$nr)
## [1] 0.7117878
GiniSimpsonMVUE(lst$nr)
## [1] 0.7117878
```

The Gini-Simpson index expresses the probability that two randomly sampled molecules from the viral population correspond to different haplotypes. It has a range of variation from 0, when there is a single haplotype, to 1 (asymptotically), when there are an infinity of equally abundant haplotypes. Interpretation of this index is clearer and more intuitive than that of Shannon entropy, and it is scarcely sensitive to rare haplotypes, as it gives higher weight to the more abundant variants.

The variance of the Gini-Simpson index is obtained with:

```
GiniSimpsonVar(lst$nr)
## [1] 0.0128987
```

- Hill numbers

Although Shannon entropy, the Gini-Simpson index, and many other reported indices are considered measures of diversity, their units do not allow an easily interpreted and intuitive measure of true diversity. They are not linear with respect to the addition of new haplotypes, and they tend to saturate: the larger the number of haplotypes, the less sensitive they are to frequency changes. In contrast, Hill numbers are a generalization of diversity measures in units of equally abundant species.

```
Hill(lst$nr,q=1)
## [1] 5.153107
```

Hill function computes the Hill number for a given order, q, but in QSutils there is a wrapper function that computes a profile of Hill numbers for different q values, so that one can see what happens when q increases in a plot.

```
HillProfile(lst$nr)
plot(HillProfile(lst$nr),type ="b", main="Hill numbers obtained
    with HillProfile")
```

```
##        q          qD
## 1   0.00 10.000000
## 2   0.10  9.364553
## 3   0.20  8.751454
## 4   0.30  8.166229
## 5   0.40  7.613843
## 6   0.50  7.098360
## 7   0.60  6.622668
## 8   0.70  6.188332
## 9   0.80  5.795587
## 10  0.90  5.443454
## 11  1.00  5.153107
## 12  1.20  4.607768
## 13  1.40  4.203660
## 14  1.60  3.891941
## 15  1.80  3.650325
## 16  2.00  3.461118
## 17  2.25  3.278284
## 18  2.50  3.138087
## 19  2.75  3.028004
## 20  3.00  2.939603
## 21  3.25  2.867153
## 22  3.50  2.806698
## 23  3.75  2.755460
## 24  4.00  2.711447
## 25  5.00  2.583506
## 26  6.00  2.501359
## 27  7.00  2.444364
## 28  8.00  2.402760
## 29  9.00  2.371234
## 30 10.00  2.346626
## 31   Inf  2.155172
```

As the order of the Hill number increases, the measure of diversity becomes less sensitive to rare haplotypes. Note that the Hill number of order $q=0$ is simply the number of haplotypes. For q=1 the Hill number is undefined, but as approaches 1 it tends to the exponential of Shannon entropy. For $q=2$ the Hill number corresponds to the inverse of the Simpson index and for $q=$`Inf` it is the inverse of the relative abundance of the rarest haplotype.

- Rényi entropy

```
Renyi(lst$nr,q=3)
## [1] 1.078274
```

Rényi entropy is a log transformation of the Hill numbers.

```
RenyiProfile(lst$nr)
plot(RenyiProfile(lst$nr),type ="b", main="Rényi entropy obtained with
    RenyiProfile")
```

```
##        q      renyi
## 1   0.00 2.3025851
```

**Hill numbers obtained
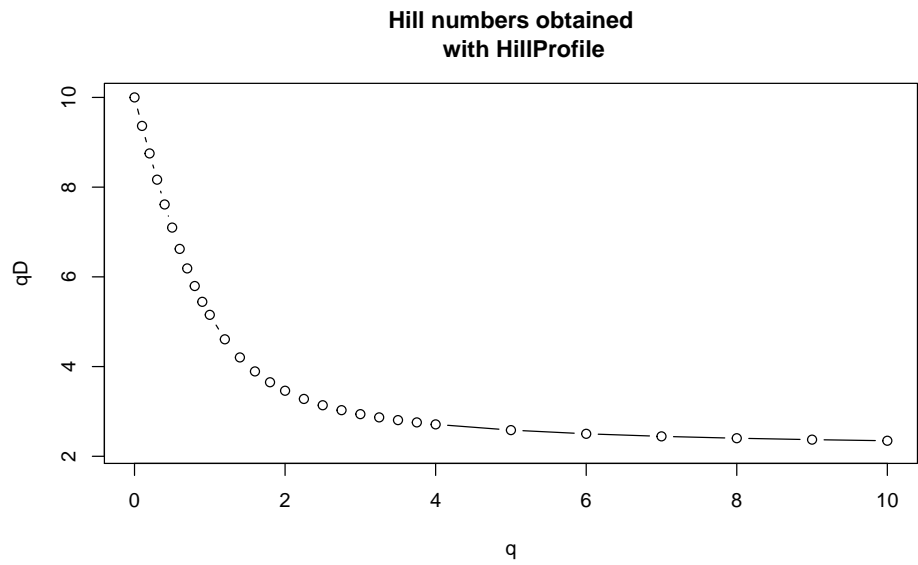with HillProfile**

Figure 1: **Hill numbers profile**

```
## 2    0.10 2.2369316
## 3    0.20 2.1692199
## 4    0.30 2.1000072
## 5    0.40 2.0299680
## 6    0.50 1.9598637
## 7    0.60 1.8904982
## 8    0.70 1.8226656
## 9    0.80 1.7570967
## 10   0.90 1.6944139
## 11   1.00 1.6395998
## 12   1.20 1.5277437
## 13   1.40 1.4359555
## 14   1.60 1.3589079
## 15   1.80 1.2948162
## 16   2.00 1.2415916
## 17   2.25 1.1873200
## 18   2.50 1.1436134
## 19   2.75 1.1079036
## 20   3.00 1.0782744
## 21   3.25 1.0533194
## 22   3.50 1.0320087
## 23   3.75 1.0135843
## 24   4.00 0.9974824
## 25   5.00 0.9491472
## 26   6.00 0.9168340
## 27   7.00 0.8937851
## 28   8.00 0.8766183
## 29   9.00 0.8634104
## 30  10.00 0.8529785
## 31    Inf 0.7678707
```

**Rényi entropy obtained with
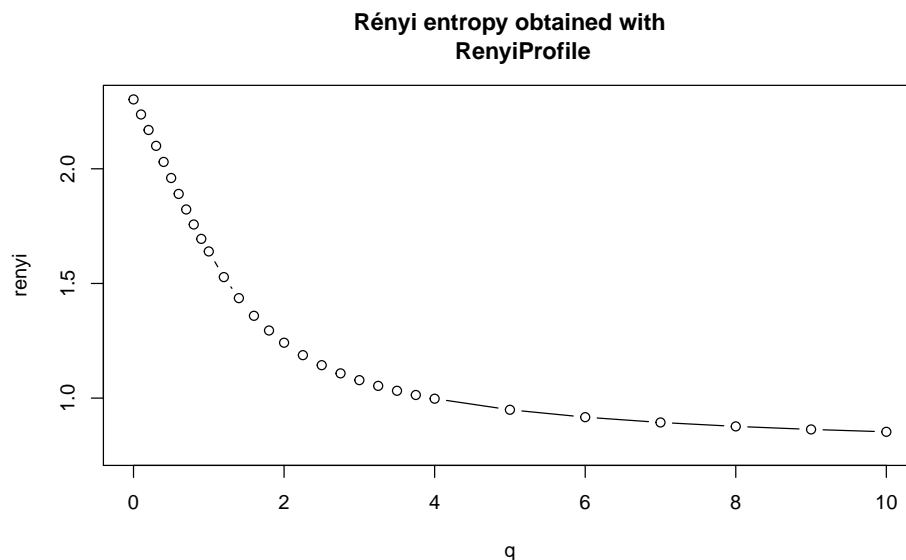RenyiProfile**

Figure 2: Rényi entropy profile

`RenyProfile`, a wrapper function of `Renyi`, computes the Rényi entropy for a set of predefined q to compute a plot to see the behavior of this index as q increases.

- Havrda-Charvat estimator

```
HCq(lst$nr, q= 4)
## [1] 0.3166118
```

Havrda-Charvat entropy, a measure of the diversity of each population, is a generalization of Shannon entropy.

```
HCqVar(lst$nr, q= 4)
## [1] 3.860805e-06
```

The asymptotic variance for a given exponent of this estimator can be computed with `HCqVar`.

```
HCqProfile(lst$nr)
plot(HCqProfile(lst$nr),type ="b", main="Havrda-Charvat entropy obtained
    with HCqProfile")
```

```
##         q         HC
## 1    0.00 9.0000000
## 2    0.10 7.2083625
## 3    0.20 5.8388239
## 4    0.30 4.7846529
## 5    0.40 3.9672588
## 6    0.50 3.3285494
## 7    0.60 2.8254117
## 8    0.70 2.4257164
## 9    0.80 2.1054156
## 10   0.90 1.8464291
```

```
## 11  1.00 1.6395998
## 12  1.20 1.3164050
## 13  1.40 1.0923685
## 14  1.60 0.9291887
## 15  1.80 0.8063456
## 16  2.00 0.7110760
## 17  2.25 0.6186440
## 18  2.50 0.5467412
## 19  2.75 0.4892160
## 20  3.00 0.4421381
## 21  3.25 0.4028962
## 22  3.50 0.3696911
## 23  3.75 0.3412423
## 24  4.00 0.3166118
## 25  5.00 0.2443882
## 26  6.00 0.1979576
## 27  7.00 0.1658853
## 28  8.00 0.1425482
## 29  9.00 0.1248749
## 30 10.00 0.1110596
## 31   Inf 0.0000000
```
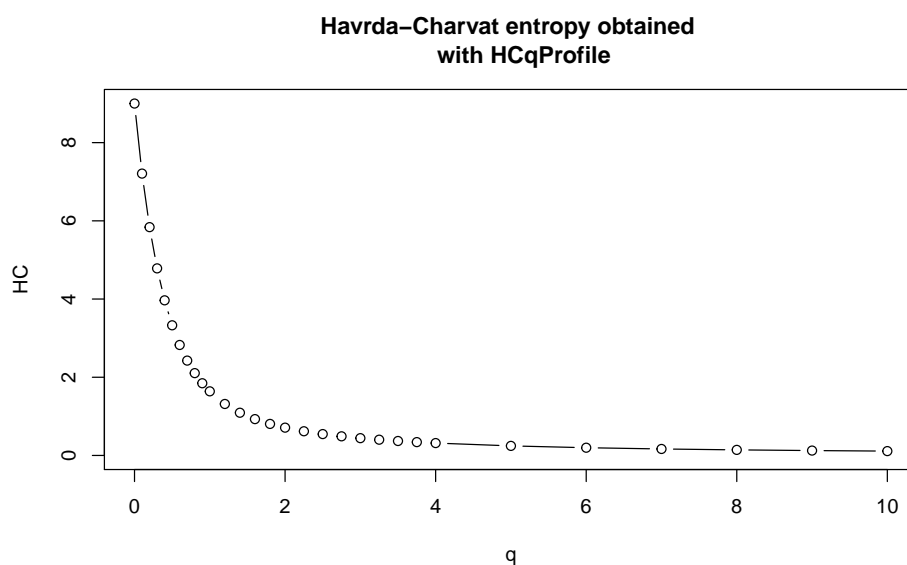

Figure 3: Havrda-Charvat entropy profile

The `HCq` function computes Havrda-Charvat entropy for a given order, q, and HCqProfile is a wrapper function that computes a profile of Havrda-Charvat entropy for different q values.

5   Functional diversity

Going a step further, functional diversity takes into account the differences between haplotypes in the quasispecies by considering their genetic distances.

5.1 Incidence-based functional diversity indices

This set of functions only considers the distances between haplotypes.

```
dst <- DNA.dist(lst$hseqs,model="raw")
library(psych)
D <- as.matrix(dst)
rownames(D) <- sapply(rownames(D),function(str) strsplit(str,
            split="\\|")[[1]][1])
colnames(D) <- rownames(D)
D
##            Hpl_0_0001 Hpl_1_0001 Hpl_1_0002 Hpl_1_0003 Hpl_2_0001
## Hpl_0_0001       0.00       0.02       0.02       0.02       0.04
## Hpl_1_0001       0.02       0.00       0.04       0.04       0.02
## Hpl_1_0002       0.02       0.04       0.00       0.04       0.06
## Hpl_1_0003       0.02       0.04       0.04       0.00       0.06
## Hpl_2_0001       0.04       0.02       0.06       0.06       0.00
## Hpl_4_0001       0.08       0.06       0.10       0.10       0.08
## Hpl_5_0001       0.10       0.08       0.12       0.12       0.10
## Hpl_8_0001       0.16       0.14       0.18       0.18       0.16
## Hpl_9_0001       0.18       0.16       0.20       0.16       0.18
## Hpl_10_0001      0.20       0.18       0.22       0.18       0.20
##            Hpl_4_0001 Hpl_5_0001 Hpl_8_0001 Hpl_9_0001 Hpl_10_0001
## Hpl_0_0001       0.08       0.10       0.16       0.18        0.20
## Hpl_1_0001       0.06       0.08       0.14       0.16        0.18
## Hpl_1_0002       0.10       0.12       0.18       0.20        0.22
## Hpl_1_0003       0.10       0.12       0.18       0.16        0.18
## Hpl_2_0001       0.08       0.10       0.16       0.18        0.20
## Hpl_4_0001       0.00       0.10       0.16       0.18        0.20
## Hpl_5_0001       0.10       0.00       0.06       0.08        0.10
## Hpl_8_0001       0.16       0.06       0.00       0.02        0.04
## Hpl_9_0001       0.18       0.08       0.02       0.00        0.02
## Hpl_10_0001      0.20       0.10       0.04       0.02        0.00
```

- Average mutation frequency by entity

```
MutationFreq(dst)
## [1] 0.082
```

The mean mutation frequency by entity, *Mfe*, measures the fraction of nucleotides in the haplotype alignment that differ from the dominant haplotype in the quasispecies

- FAD

```
FAD(dst)
## [1] 9.88
```

The functional attribute diversity, *FAD*, is the sum of pairwise distances between haplotypes in the alignment.

- Nucleotide diversity by entity

```
NucleotideDiversity(dst)
##           [,1]
## [1,] 0.1097778
```

The nucleotide diversity by entity, π_e, is a transformation of `FAD` and it expresses the average difference between haplotypes in the alignment.

5.2   Abundance-based functional diversity indices

- Average mutation frequency by molecule

```
nm <- nmismatch(pairwiseAlignment(lst$hseqs,lst$hseqs[1]))
MutationFreq(nm,lst$nr,len=width(lst$hseqs)[1])
## [1] 0.0659
```

The proportion of different nucleotides at the molecular level, *Mfm*, measures the fraction of nucleotides in the population that differ from the dominant haplotype in the quasispecies.

```
MutationFreqVar(nm,lst$nr,len=width(lst$hseqs)[1])
## [1] 6.41759e-06
```

The variance of *Mfm* is calculated with the `MutationFreqVar` function.

- Nucleotide diversity

```
NucleotideDiversity(dst,lst$nr)
##           [,1]
## [1,] 0.08634655
```

The nucleotide diversity, π_m, which is related to Rao entropy in ecology, corresponds to the mean genetic distance between molecules in the population.

- Rao entropy

```
RaoPow(dst,4,lst$nr)
## [1] 6.332901e-05
```

To obtain the Rao entropy for a given order of *q*, the RaoPow function is used.

```
RaoVar(dst,lst$nr)
##             [,1]
## [1,] 2.345394e-06
```

The variance of π_m is calculated with the `RaoVar` function.

```
RaoPowProfile(dst,lst$nr)
plot(RaoPowProfile(dst,lst$nr),type ="b", main="Rao entropy obtained
    with RaoPowProfile")
```

```
##      q          qQ
## 1  0.0 9.880000000
## 2  0.1 5.494606337
```

```
## 3  0.2 3.124207222
## 4  0.3 1.819585453
## 5  0.4 1.087184512
## 6  0.5 0.667121028
## 7  0.6 0.420633660
## 8  0.7 0.272493135
## 9  0.8 0.181233347
## 10 0.9 0.123589240
## 11 1.0 0.086260200
## 12 1.1 0.061491451
## 13 1.2 0.044668953
## 14 1.3 0.032989839
## 15 1.4 0.024715352
## 16 1.5 0.018743915
## 17 1.6 0.014362892
## 18 1.7 0.011101667
## 19 1.8 0.008643155
## 20 1.9 0.006769532
## 21 2.0 0.005328380
```
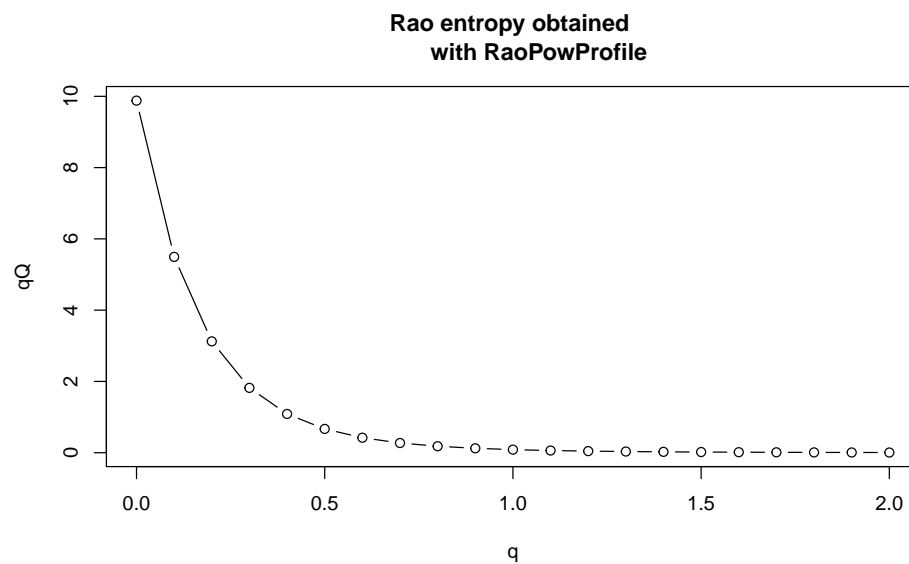

Figure 4: **Rao entropy profile**

6   Sample size and bias

The diversity we measure in viral quasispecies samples is necessarily tied to the sample size. The larger the coverage, the larger the number of haplotypes, polymorphic sites, and mutations we can find. The same is true for Shannon entropy and other diversity indices. Because of the variability in all experimental procedures, the best pooling of samples never results in perfectly balanced coverage, which complicates the comparison of diversity indices biased by sample size. Hence, a sample size correction method is required to obtain fair inferences. The literature regarding ecology shows several correction methods, which unfortunately, are not

applicable to NGS because of one differential trait. With NGS, sooner or later we are obliged to eliminate all haplotypes present in frequencies below a threshold, considered the noise level of the technique. This lower end of frequencies contains the information required by most established corrections.

The correction we found that works best in this situation consists in determining the minimum coverage of all samples to be compared and down sampling the other samples to this minimum, simply by rescaling. All haplotypes with resulting frequencies below a given threshold are then removed with high confidence, in a step we have dubbed fringe trimming. The $DSFT$ function is used to correct the quasispecies composition to a given sample size. It takes four parameters: nr, the vector of haplotype frequencies in the original sample; $size$, the total number of reads in the new sample prior to abundance filtering; $p.cut$, the abundance threshold in the filter; and $conf$, the level of confidence in trimming haplotypes, which defaults to 0.95. $DSFT$ returns a vector of logicals with FALSE for all haplotypes to be removed by the correction.

The next code shows the impact of each data treatment step when determining Shannon entropy. First, we simulate the frequencies of haplotypes in a quasispecies population. Then we take repeated samples from this population, 2000 and 5000 reads in size. We then compute Shannon entropy of the raw samples,and the samples after noise filtering at an abundance of 0.1%, and after the DSFT corrections at size 2000, and filtering at 0.2% with 95% confidence.

```r
set.seed(123)
n <- 2000
y <- geom.series(n,0.8)+geom.series(n,0.00025)
nr.pop <- round(y*1e7)

thr <- 0.1
sz1 <- 5000
sz2 <- 10000

qs.sample <- function(nr.pop,sz1,sz2){
    div <- numeric(6)
    nr.sz1 <- table(sample(length(nr.pop),size=sz1,replace=TRUE,p=nr.pop))
    div[1] <- Shannon(nr.sz1)
    nr.sz1 <- nr.sz1[nr.sz1>=sz1*thr/100]
    div[3] <- Shannon(nr.sz1)
    div[5] <- Shannon(nr.sz1[DSFT(nr.sz1,sz1)])

    nr.sz2 <- table(sample(length(nr.pop),size=sz2,replace=TRUE,p=nr.pop))
    div[2] <- Shannon(nr.sz2)
    nr.sz2 <- nr.sz2[nr.sz2>=sz2*thr/100]
    div[4] <- Shannon(nr.sz2)
    div[6] <- Shannon(nr.sz2[DSFT(nr.sz2,sz1)])
    div
}

hpl.sim <- replicate(2000,qs.sample(nr.pop,sz1,sz2))
nms <- paste(c(sz1,sz2))

par(mfrow=c(1,3))

boxplot(t(hpl.sim[1:2,]),names=nms,col="lavender",las=2,
```

```
            ylab="Shannon entropy",main="raw")
boxplot(t(hpl.sim[3:4,]),names=nms,col="lavender",las=2,
            ylab="Shannon entropy",main="filt")
boxplot(t(hpl.sim[5:6,]),names=nms,col="lavender",las=2,
            ylab="Shannon entropy",main="DSFT")
```
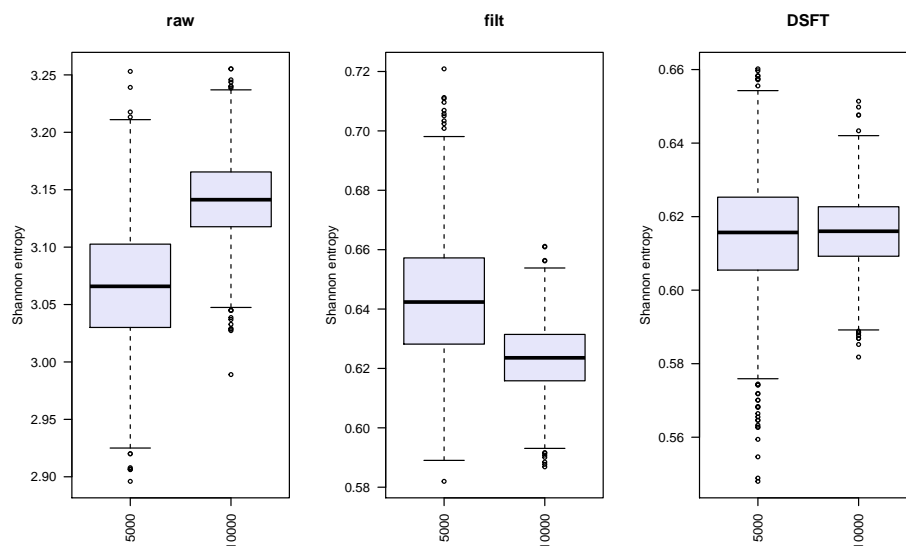

Figure 5: Diversity index variations due to sample size

As is evident, the distribution of Shannon entropy values in raw samples of 10000 reads shows higher values than those in samples of 5000 reads, even though the samples come from the same viral population.

After removing all haplotypes with frequencies below 0.1%, the effect is reversed: Shannon entropies are much lower for the larger samples. Correction by DSFT brings the two distributions to comparable levels.

This behavior is typical of samples with long queues of very low fitness and defective haplotypes, possibly worsened by technical errors in sample preparation and sequencing.

6.1   The load of rare haplotypes

We found that the fraction of reads belonging to haplotypes below 1% in frequency, the rare haplotype load (RHL), is a robust index of quasispecies diversity, especially suitable for detecting mutagenic processes. This index is also robust and does not require sample size corrections, provided that coverage is high enough. The RHL computation is done without a previous abundance filter.

Let's generate a random quasispecies with an RHL of about 25%:

```
set.seed(123)
n <- 2000
y <- geom.series(n,0.8)+geom.series(n,0.0002)
nr.pop <- round(y*1e7)
rare <- nr.pop/sum(nr.pop) < 0.01
```

```
RHL <- sum(nr.pop[rare])/sum(nr.pop)
round(RHL,4)
## [1] 0.2535
```

Repeating the sampling process seen before for Shannon entropy, we obtain median RHL values that are very close to the population value for both sample sizes.

The population value is shown in the next figure as a dash-dot horizontal line.

```
thr <- 0.1
sz1 <- 5000
sz2 <- 10000
qs.sample <- function(nr.pop,sz1,sz2){
    div <- numeric(2)
    nr.sz1 <- table(sample(length(nr.pop),size=sz1,replace=TRUE,p=nr.pop))
    rare <- nr.sz1/sum(nr.sz1) < 0.01
    div[1] <- sum(nr.sz1[rare])/sum(nr.sz1)
    nr.sz2 <- table(sample(length(nr.pop),size=sz2,replace=TRUE,p=nr.pop))
    rare <- nr.sz1/sum(nr.sz2) < 0.01
    div[2] <- sum(nr.sz2[rare])/sum(nr.sz2)
    div
}

hpl.sim <- replicate(2000,qs.sample(nr.pop,sz1,sz2))
nms <- paste(c(sz1,sz2))
boxplot(t(hpl.sim),names=nms,col="lavender",las=2,ylab="RHL")
abline(h=RHL,lty=4,col="navy")
```
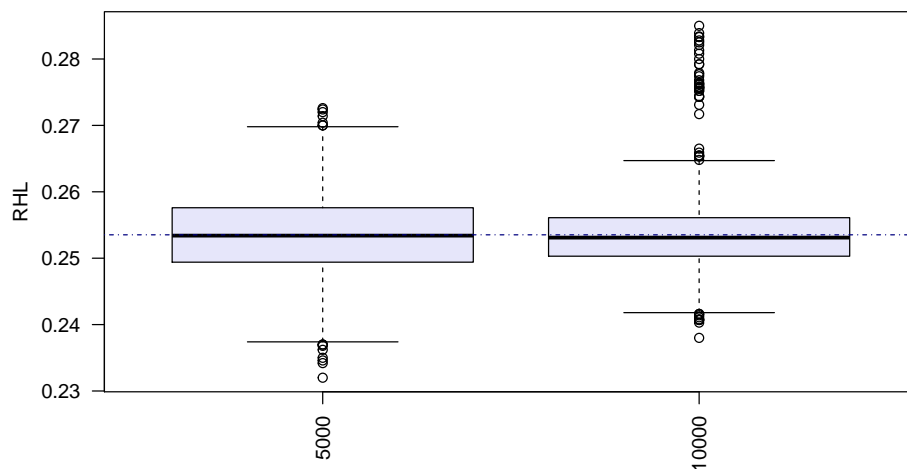


```
## R version 3.4.4 (2018-03-15)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 18.04.1 LTS
##
## Matrix products: default
## BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.7.1
## LAPACK: /usr/lib/lapack/liblapack.so.3.6.0
```

```
##
## locale:
##  [1] LC_CTYPE=ca_ES.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=ca_ES.UTF-8        LC_COLLATE=ca_ES.UTF-8
##  [5] LC_MONETARY=ca_ES.UTF-8    LC_MESSAGES=ca_ES.UTF-8
##  [7] LC_PAPER=ca_ES.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=ca_ES.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats4    parallel  stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
## [1] psych_1.8.4      QSutils_0.99.0     Biostrings_2.46.0
## [4] XVector_0.18.0   IRanges_2.12.0     S4Vectors_0.16.0
## [7] BiocGenerics_0.24.0 devtools_1.13.6  BiocStyle_2.6.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.17    knitr_1.20      magrittr_1.5    zlibbioc_1.24.0
##  [5] mnormt_1.5-5    ape_5.1         lattice_0.20-35 stringr_1.3.1
##  [9] tools_3.4.4     grid_3.4.4      nlme_3.1-137    xfun_0.3
## [13] git2r_0.21.0    withr_2.1.2     htmltools_0.3.6 yaml_2.2.0
## [17] rprojroot_1.3-2 digest_0.6.15   bookdown_0.7.17 memoise_1.1.0
## [21] evaluate_0.10.1 rmarkdown_1.10  stringi_1.2.3   compiler_3.4.4
## [25] backports_1.1.2 foreign_0.8-70
```

References

Gregori, Josep, Celia Perales, Francisco Rodriguez-Frias, Juan I. Esteban, Josep Quer, and Esteban Domingo. 2016. "Viral quasispecies complexity measures." *Virology* 493. Elsevier: 227–37. doi:10.1016/j.virol.2016.03.017.

Gregori, Josep, Miquel Salicrú, Esteban Domingo, Alex Sanchez, Juan I. Esteban, Francisco Rodríguez-Frías, and Josep Quer. 2014. "Inference with viral quasispecies diversity indices: Clonal and NGS approaches." *Bioinformatics* 30 (8): 1104–11. doi:10.1093/bioinformatics/btt768.